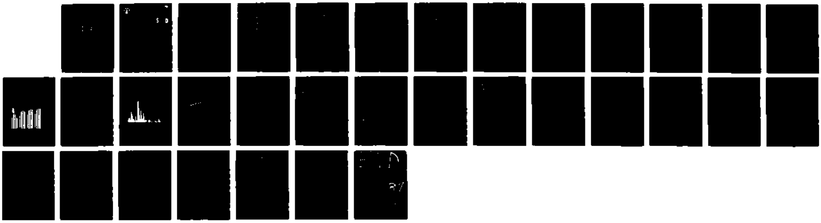
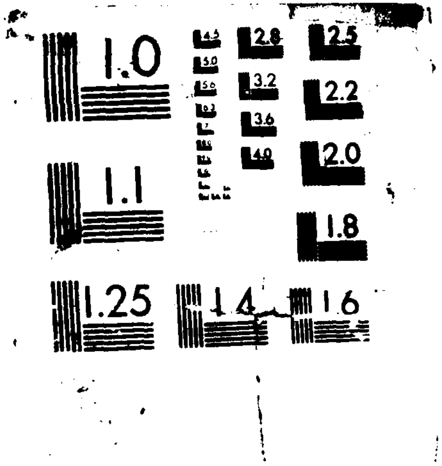
AD-A184 089   RESEARCH IN VLSI (VERY LARGE SCALE INTEGRATION) SYSTEMS   1/1
(U) STANFORD UNIV CA COMPUTER SYSTEMS LAB
J HENNESSY ET AL. MAR 87 MDA903-84-K-0062

UNCLASSIFIED   F/G 9/1   NL





MICROCOPY RESOLUTION TEST CHART

1

COMPUTER SYSTEMS LABORATORY
STANFORD UNIVERSITY · STANFORD, CA 94305-2192

DTIC FILE COPY

AD-A184 089

Research in VLSI Systems

DTIC
ELECTE
AUG 2 7 1987
S D
D

Technical Progress Report

December 1986 - March 1987

Computer Systems Laboratory
Integrated Circuits Laboratory
Center for Integrated Systems

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

This work was supported by the Defense Advanced Research Projects Agency, contracts MDA903-84-K-0062 and MDA903-83-C-0335.

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

87 8 17 005

Department of Electrical Engineering • Department of Computer Science

87-3689

Research in VLSI Systems

Progress Report for December 1986 - March 1987

Center for Integrated Systems
Stanford University
Stanford, California 94305

General Purpose VLSI-Based Multiprocessors
DARPA Contract No. MDA903-83-C-0335
DARPA Order No. 3773-6
Principal Investigator: John Hennessy
Computer Systems Laboratory
Monitored by W. Bandy

A Fast Turn Around Facility for Very Large Scale Integration (VLSI)
DARPA Contract Nos. MDA903-80-C-0432 and MDA903-84-K-0062
Principal Investigator: James Plummer
Integrated Circuits Laboratory
Monitored by W. Bandy

Accesion For
NTIS CRA&I ☑
DTIC TAB □
Unannounced □
Justification
By per ltr.
Distribution /
Availability Codes
Dist | Avail and / or Special
A-1

DTIC
COPY
INSPECTED
6

Table of Contents

List of Figures

Abstract

This report summarizes progress in the DARPA funded VLSI Systems Research Projects from December 1986 to March 1987. The major areas under investigation have included: analysis and synthesis design aids, applications of VLSI, special purpose chip design, VLSI computer architectures, reliability studies, manufacturing science, and VLSI fabrication. The major research problems are introduced and progress is discussed; the Appendix contains a list of published research papers from these projects.

Key Words and Phrases: VLSI, design automation, computer-aided design, special purpose chips, VLSI computer architecture, routing, layout, memory reliability, manufacturing science, IC fabrication.

This work was supported by the Defense Advanced Research Projects Agency, contracts MDA903-84-K-0062 and MDA903-83-C-0335.

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

Executive Summary

The major progress of note for this period is as follows:

1. *MIPS-X: a very high performance VLSI processor.* MIPS-X [Chow 86, Chow 85, Horowitz 87, Chow 87] is a project to develop a very high performance processor to be used as the node processor in a high performance multiprocessor. Like MIPS, MIPS-X uses a simplified instruction set, a deep pipeline, and code reorganization to increase performance. Unlike MIPS, MIPS-X contains an on chip instruction cache, and supports both coprocessors and a multiprocessor environment. First silicon on MIPS-X is fully functional with parts operating up to 17 MHz (with a target of 20 MHz). A system test board has been designed and is currently in PC-board layout. On going work is focused on performance improvements and a shrink to 1.25 μ.

2. *High Speed Multiplication and Division.* Two chips have been designed to test new ways to implement the components of a high performance floating point processor. Both of the chips use a small array of elements and iterate around the array. The divison chip has been fabricated and runs at 13ns/quotient bit [Williams 87]. The multiplication chip is currently in fabrication, we expect it out the end of April.

3. *Software support for RISC processors.* We have continued to explore methods of improving the effective performance of a processor by improving the quality of the code generated by the software system. Recent work has focused on new interprocedural analysis algorithms and on efficient implementation of LISP. The LISP efforts have studied tags and a software register window scheme; together this optimizations significantly improve the performance of LISP without the addition of any hardware support.

4. *Automatic partitioning of parallel programs.* A system for partitioning dataflow graphs into multiple tasks for execution on a parallel processor has been developed. Current efforts are focused on a port of the system to a commercial multiprocessor (the Encore). Related work has focused on optimization problems in the functional languages that generate our data flow graphs, and a new technique for copy elimination has been devised.

5. *RSIM.* We have continued our work on improving the models used in switch level simulation. By using a simple two timeconstant model the effects of charge-sharing can be naturally folded into the node evalution [Chu 86]. This model has been extended to handle transistor-capacitor circuits. Although the nonlinearity prevents a true two timeconstant model, one can still reduce the circuit into a cononic form, and use a table of precomputed values for the solution.

6. *Resistance Extraction.* To obtain more accurate delay estimates in integrated circuits we have integrated a resistance extractor into the Magic layout system [Stark 87]. This extractor uses a simple square counting algorithm to determine a wire's resistance. It also uses a series of filters and simplifying routines to only create resistors that have a significant effect on the circuits performance. We have successfully run this program on the MIPS-X database.

7. *THOR: A Functional Simulator.* The Thor system integrates RSIM, the medium tester, and a functional simulator into one environment allowing easy consistency

checking between different representations. Also included in the system is CSLIM which takes a simple behavioral description and generates PLA equations, and a logic analyzer for viewing the simulation waveforms. The software is ready for beta testing outside Stanford.

8. *Testing Chip* In conjunction with MOSIS, we have designed a special purpose memory chip that will enable us to build a high speed tester at a low cost [Miyamoto 87]. The chip acts as a small test vector memory and a set of very flexible input output pads. The 3μ version of the chip runs over 10MVector/sec and the 2μ version of the chip run over 16MVectors/sec.

9. *Testable CMOS Design*. Testable circuit structures have been developed that can be used to design easily testable CMOS VLSI circuits. One structure is a built-in self-test PLA, and the other structure is a self- testable application-specific IC (ASIC). These structures contain circuitry to generate test patterns as well as to evalute test responses. Hence, they can reduce the complexity of IC testing and the dependence on high-cost testers.

10. *Computer Support — Fable*. We have initiated a course entitled "Automation of Semiconductor Manufacturing" which is bringing together AI and wafer fabrication experts to attack several problems of importance to the Computer Automated Fabrication effort. These groups are working in an advanced TI Explorer/KEE environment.

11. *Computer Integrated Manufacturing e-mail discussion group*. A moderated inter-university news group has been established to discuss matters of interest to the Computer Automated Fabrication community. Join by sending your net address to IC-CIM-Request@Sierra.Stanford.EDU

12. *Electrical alignment test structures*. A comprehensive set of test structures which monitor ΔX and ΔY registration accuracy have been developed.

13. *Template-set matching for random defect detection*. A 2 μm CMOS circuit has been designed to aid in random defect inspection of masks and integrated circuits. A template-set matching scheme has been applied to the task of defect detection and, more recently, to defect classification.

Technical Progress

1 Design Description, Analysis, and Synthesis

1.1 Circuit Modeling for Simulation

We have continued our work on improving the models that are used in switch level simulation. Our work in this area is based on the RSIM simulator from MIT. Our recent work has concentrated on using a two time-constant model to improve the timing model and charge sharing model in switch level simulation. This model was originally derived for linear networks and has been extended so it can model transistor capacitor circuits. These circuit are first reduced into a cononical two-transistor two-capacitor circuit, which is characterized by only two parameters. The small number of variables (2) allows one to presolve the problem and store the results in a table form if that is required. Using this method we have determined that transistor-capacitor circuits are always less susceptible to voltage spikes than RC networks.

We have also been working on the algorithm used to find new node values in the simulator to try to understand and fix the EXOR problem that affects all simulators. This problem arises because the simulator decouples the value on a transistors gate while it finds the new source drain values. To avoid the decoupling problem, we map transistors with self connected gates into a MOS diode and try to solve the network. When the solution is found we check our assumptions about the transistors operation. If there were incorrect we solve the network again. We are adding this algorithm to the RSIM simulator.

Staff: C.Y. Chu, M Horowitz

Related Efforts: COSMOS (CMU)

References: [Chu 86]

1.2 Resistance Extraction

Parasitic resistances can substantially affect circuit performance but are difficult to calculate efficiently. We have implemented an extractor designed to produce resistance values for use in digital circuit simulation. The extracter begins with the crude resistance values that are provided by the Magic extractor. These values are used as a filter to select nodes that might have significant resistances. For each of the node that could be a problem, the extractor first finds an approximation of the resistance value by using a simple squares counting algorithm. The time constant of the wire is compared against the time constant of the driver, and if the wire delay is under a tolerance, the resistance is ignored. If the wire delay is significant, the resistance network is simplified to reduce the number or resistors and nodes needed to model the network and then output into a file. The REDS extractor has been run on the MIPS-X database, and required about 2 CPU hours on a uVax to complete. The lumped resistance (Magic resistance values) filter was effective in reducing the number of nodes that needed to be extracted; only 20% failed. After extraction only 0.4% of the nodes failed (83), most of these were in the pads (58). The pads are not really a problem, they are flaged because without the external load the time-constant of the output stage is extremely small. We were quite pleased that REDS did not find any long wire delays that were not already known.

Staff: D. Stark, M Horowitz

References: [Stark 87]

1.3 Thor Simulation System

The THOR research is broken into three major areas: a production functional simulation environment, incremental simulation research, and parallel simulation research. The simulation environment is a collection of tools that can be used either for VLSI chip simulation or systems simulation. The incremental research examines trade-offs for tracking changes through a design process. The parallel simulation research examines the tradeoffs between different representation levels and the total available parallelism. Each effort is discussed separately in the following sections.

1.3.1 THOR Environment

While a true mixed-level simulation has some advantages, it is not the best method for simply verifying a design. For this, it is more useful to simulate the two different levels of design in parallel and check for discrepancies. This forces a close correspondence between the different levels of design, which aids the whole design process. The THOR functional simulation systems provides this capability. Specifically, interfaces are provided for RSIM, the medium chip tester, a logic analyzer, and a state machine design synthesis tool (CSLIM – described later). THOR provides an integrated way that functional simulation can be verified against the extracted switch simulation and against the physical chip. Using this approach, test vectors are easily generated from the functional simulation and are used to stimulate the lower design abstractions.

CSLIM generates PLA equations in espresso format from a THOR behavioral model. The underlying idea of CSLIM is to generate the PLA for layout with the same model that is used in the simulation, to reduce transcription errors and to shorten the debug cycle. The input is a restricted THOR model that can use any combination of if, switch, and EXITMOD control constructs, assignment statements, and boolean expressions. CSLIM analyzes the model and calculates a set of logic equations describing the inputs in terms of the output. It generates a full don't care set for maximum minimization. It also checks that every output is assigned on every execution of the model, so false state on the outputs introduced by the nature of the simulator does not affect the functioning of the PLA. CSLIM does not do extensive logic minimization; it relies on espresso to do this for it. CSLIM allows more general control structures than existing PLA generators while integrating the simulation and PLA generation into one system.

The logic analyzer displays the state of a simulation in a graphical, easy to understand, way and may be run in real-time, in parallel with the simulation, or in 'batch'; where the results of a previous run may be inspected. The analyzer provides a convenient way for looking at arbitrary groups of signals (buses) in (user-programmable) numerical bases. Undefined signals and "glitches" are easily identifiable, easing system debugging. Commands allow the user to easily move back, forward, zoom-in, and zoom-out in time. Also provided are commands that allow tracing of one or more signals on various conditions such as equality, inequality, change, etc. Finally, hardcopy is possible of any waveform display.

1.3.2 Incremental Simulation

The incremental simulator is based on the observation that most design changes have relatively small implications and the effects can be computed very quickly. It adopts its own node/element scheduling and event propagation mechanisms. While a conventional event-driven selective-trace simulator starts simulation from the input stimuli over the entire circuit, the incremental simulator simulates only the circuit components affected by the network changes. In the incremental simulation, it is not the circuit size but the implications of the circuit changes, i.e., fanouts of the net nodes whose connections have been changed, that determines the simulation time.

An incremental simulator has been proposed and implemented. We started gathering statistics for the implemented program. Preliminary results show that speedups between three and seven can be obtained when simulating incrementally for minor changes on our test circuits. Further tests on larger examples are required.

1.3.3 Parallel Simulation Study

Two basic areas have been studied for increasing system simulation performance: abstraction level and maximum available parallism. In the paper "Statistics for Parallelism and Abstraction Level in Digital Simulation", (accepted for the 24th Design Automation Conference) we evaluate the performance implications of different design representation levels and found roughly a 10X speed-up between each of the levels.

Attacking parallel simulation, we have the THOR simulator running on an Encore and have achieved utilization factors of 70-80% using 4-6 processors. With more processors, simulations show that parallelism can achieve speed-ups between 10-30. This work has only used static data partitions and the complete simulation algorithm on each node. Future work will examine more dynamic strategies and separating the simulation algorithm across multiple processors.

Staff: B. Alverson, S.Y. Hwang, L. Soule, T. Rokicki, K.Y. Choi, A. Salz and T. Blank

Related Efforts: THOR functional simulation language based on CSIM from the University of Colorado

1.4 Physical Placement

The main development of the automatic placement tool are the refinement of the analytical model and further examination of the numerical techniques most suitable for the placement problem. Specifically, the analytical model calculating the objective function has been improved to include actual pin positions and to allow mirror operations. The model for the objective function is now complete in the sense that all the layout operations -- translation, rotation and reflection have been included. Comparison with the results of the original model has shown that a significant reduction in the modified wire length can be achieved with above improvements.

In the efforts to develop a more efficient solution technique, the early non-linear programming technique -- a penalty function method has been compared with the widely used sequential quadratic programming method. The penalty function has been shown to be better by all accounts. Through the cooperation with the numerical optimization group in operational research department at Stanford, the plausible reasons for the seemingly strange results are understood and a potentially more efficient solution technique has been identified which involves using a quasi-Newton technique tailored to the special characteristics of the placement problem. The algorithm has been tested on some industrial examples (the biggest one has 33 blocks and 121 signal nets) showing promising results.

Staff: L. Sha and T. Blank

Related Efforts: Timberwolf at Berkeley

2 VLSI Processor Architecture and Software

2.1 MIPS-X: A High Performance VLSI Computer

The MIPS-X uniprocessor design goal is a machine with a 20 MIP peak instruction rate, and an 'average' throughput of over 10 MIPs. The architecture is of the reduced instruction set variety, but also ventures into two new and important areas:

1. supporting high performance co-processors, and
2. providing the capability to be used in a medium-scale multiprocessor environment.

In addition, we have several closely related activities. These involve studying the implementation of LISP on MIPS-X, and the performance and analysis of very large caches.

2.1.1 Hardware Status

During this period we have received completely functional MIPS-X processor chips, and are now focusing our attention on using these chips in board level systems. While working on the board designs we noticed a few places where a small change to the external interface of the chip would make the board design much simpler. After looking these changes over we decided to include them on the next revision of the chip along with some performance improvements. The major change in the interface deals with the time the store data is presented to the memory system. The current design presents this data early and forces the board to include a set of latches on the data bus. By delaying the data for a cycle we can eliminate the need for the latch, and remove a nasty critical path from the board. The logic for this change has already been designed and we are currently in layout. We also plan to speed up the slowest paths in the machine so this version should operate at over 20MHz. This involves only a small amount redesign, and we hope to have the revised part for fabrication by the end of Spring Quarter.

The first test board for the MIPS-X processor has been designed and is in PCB layout. We are working with Bob Parker at ISI and he is using this board to evaluate different PCB layout systems. The board is very simple. It contains a clock generator, slave VME bus controller, and 64K bytes of fast static rams. The board will plug into a SUN workstation, and the memory can be read or written from the host. We should be able to download programs onto the board and then run them at speed. This board will allow use to do more complete performance testing of the part.

The second test board is more complex, and contains a cache to do more complete performance testing of the part. The board will contain two custom chips, the MIPS-X processor and the External Cache Processor (ECP). The functional description of the ECP has been done using

THOR, and a large portion of the layout has been completed. We are now writing a functional description of the board (we have descriptions of the two custom chips) and will uses the THOR system to test out the board before it is sent to fabrication. We expect to finish the ECP and board designs just before summer.

2.1.2 Making LISP run fast

The high-level language LISP has some features, like runtime type checking, that make it very different from C and Pascal, the two languages focused on in the design of MIPS-X. To determine which LISP operations are time critical, we measured LISP programs using a MIPS-X simulator. This data was discussed in the last progress report and published in [Steenkiste 86]. Our measurements showed that three fourths of the program execution time is used for three operations: tag handling (23%), procedure calls (26%) and stack accesses (22%). We looked at optimizations for each of these 3 time consuming operations.

We examined a wide variety of tag implementations and found that minor changes to the hardware (that would not affect clock rates) could achieve most of the benefits of full hardware type checking and tag handling. To reduce the cost of procedure calls, we first optimized and inlined a number of time critical primitive LISP operations. This speeded up the programs about 16% – half of this gain results from eliminated procedure calls. Then we merged user functions concentrating on small, non-recursive procedures (merging large or recursive had a very negative effect on instruction cache hit rates); this produced an additional 6% speedup.

The high procedure call frequency in LISP makes per-procedure register allocation less effective than in a C or Pascal environment. We implemented a simple inter-procedural register allocator that does allocation in a bottom-up order in the program call graph (similar in spirit to Wall's approach), as shown in Figure 1. As a result, different procedures use different registers, so fewer registers have to be saved across procedure calls.

This algorithm allowed us to eliminate over 70% of the stack accesses, and our 11 LISP programs ran an average of 10% faster. Recursion was the major limiting factor on the performance of the inter-procedural register allocator, although running out of registers before the top of the call graph is reached is also a difficulty. In fact, even if the register supply were not limited, recursion higher up in the call graph would limit the improvement to 2%.

Another interesting issue is how this software allocation scheme compares to hardware register windows. We explored this assuming a hardware register window scheme with 16 global registers and a variety of window schemes. This experiment showed that the hardware register window scheme required more than twice as many registers to do better than the software scheme (80 registers versus 32). This data is shown in Figure 2; the dashed line indicates the percentage of references eliminated by our software scheme.

The performance of MIPS-X for LISP look very encouraging. Although MIPS-X does not have any tagging hardware, it does have sufficient support for bitfields to handle tags efficiently. The execution of the Gabriel benchmarks on the MIPS-X simulator, which includes the effect of the (off chip) cache, show a performance that is significantly higher than the the VAX with no type checking (about 15 times faster) and also faster than a Symbolics 3600 with full type checking (about 5 times faster). Furthermore, this does not include our optimizations that further improve performance by about a third.

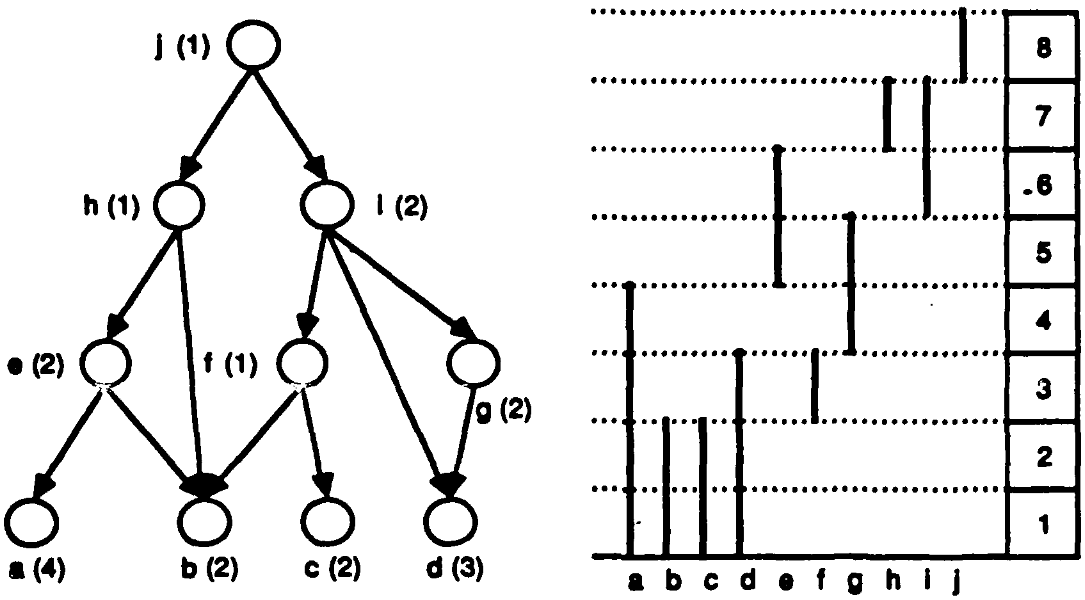


Figure 1: Interprocedural register allocation: an example

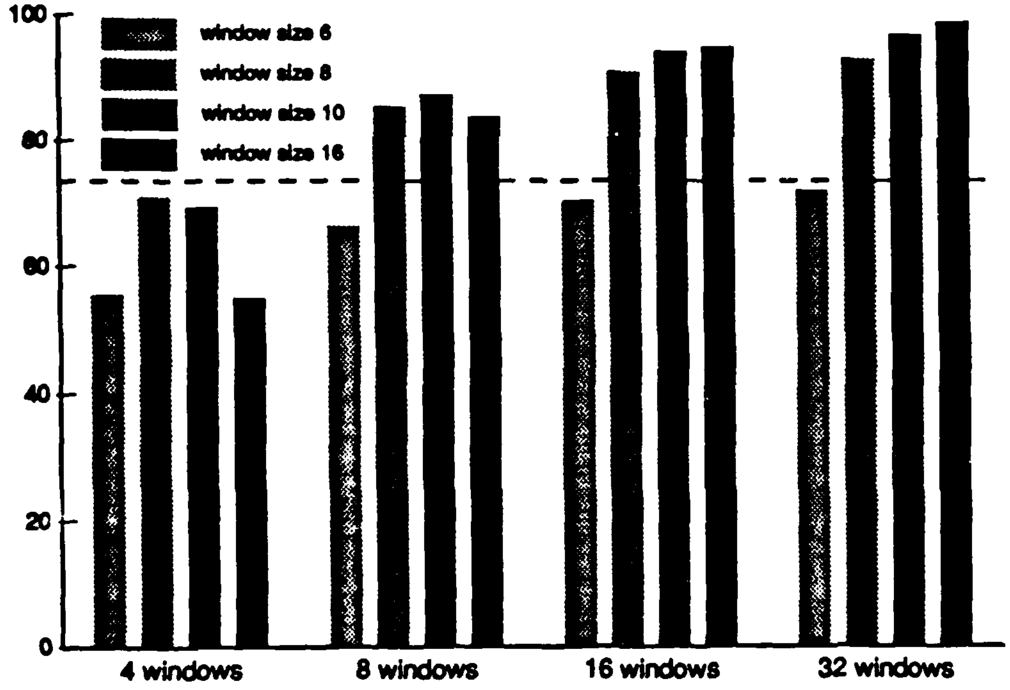


Figure 2: Percent of stack accesses removed with with register windows

2.1.3 MIPS-X Summary

Staff: S. Przybylski, C. Y. Chu, J. Hennessy, M. Horowitz, M. Wing, P. Chow, J. Acken, A. Agarwal, S. Richardson, S. McFarling, M. Ganapathi, D. Stark, R. Simoni, S. Tjiang, P. Steenkiste

Related Efforts: SPUR (Berkeley)

References: [Chow 86], [Agarwal 87] [Steenkiste 86], [Horowitz 87], [Chow 87]

2.2 Multiprocessor Support for MIPS-XMP

Our work on caches supports the MIPS-X design, but it is even more critical for our multiprocessor activities. To date the architectural work for MIPS-XMP has focused primarily on high performance memory hierarchies needed to support 8 to 10 15-mips processors; recently, we have begun work on implementing our multiprocessor architecture. We are also making progress on our software activities, which are primarily supported by an NSF grant. Since this work is an intimate part of our project we describe the results below.

2.2.1 Decomposing Parallel Programs

There are three fundamental problems to be solved in the execution of a parallel program on a multiprocessor - identifying the parallelism in the program, partitioning the program into tasks and scheduling the tasks on processors. Whereas the problem of identifying parallelism is a programming language issue, the partitioning and scheduling problems are intimately related to the number of processors and the synchronization and communication overhead in the target multiprocessor. It is desirable for the partitioning and scheduling to be performed automatically, so that the same parallel program can execute efficiently on different multiprocessors. We have investigated two solutions to the partitioning and scheduling problems. The first approach is based on a *macro-dataflow* model [Sarkar 86a], where the program is partitioned into tasks at compile-time and the tasks are scheduled on processors at run-time. The second approach is based on a *compile-time scheduling* model [Sarkar 86b], where the partitioning of the program and the scheduling of tasks on processors are both performed at compile-time.

As mentioned above, both the partitioner for macro-dataflow and the partitioner/scheduler for compile-time scheduling have already been implemented to partition SISAL programs. The partitioning is actually performed at the level of SISAL's graphical intermediate form, IF1. We extended the Livermore IF1 interpreter to produce trace files for multiprocessor simulations. We have a variety of SISAL benchmark programs, from small programs like Matrix Multiplication, Merge-exchange Sort, FFT (approximately 100 lines each) to larger programs like SIMPLE and SLAB (approximately 2000 lines each). As an example of some of the data produced, Figure 3 shows the parallelism profile for the SIMPLE benchmark. The peak parallelism (at the level of primitive operators such as an add is 1400); the average parallelism at this low granularity level is 125. A typical shared memory multiprocessor can exploit about 5 processors, due to communication and scheduling costs for this small problem size. Figure 4 shows typical speedup curves for some benchmarks.

The goal of our project is to make programs run efficiently on a wide variety of architectures with the compiler dealing with the architectural differences. Our next step will be to complete a sequential implementation and then port the system to a real multiprocessor; work is preceding

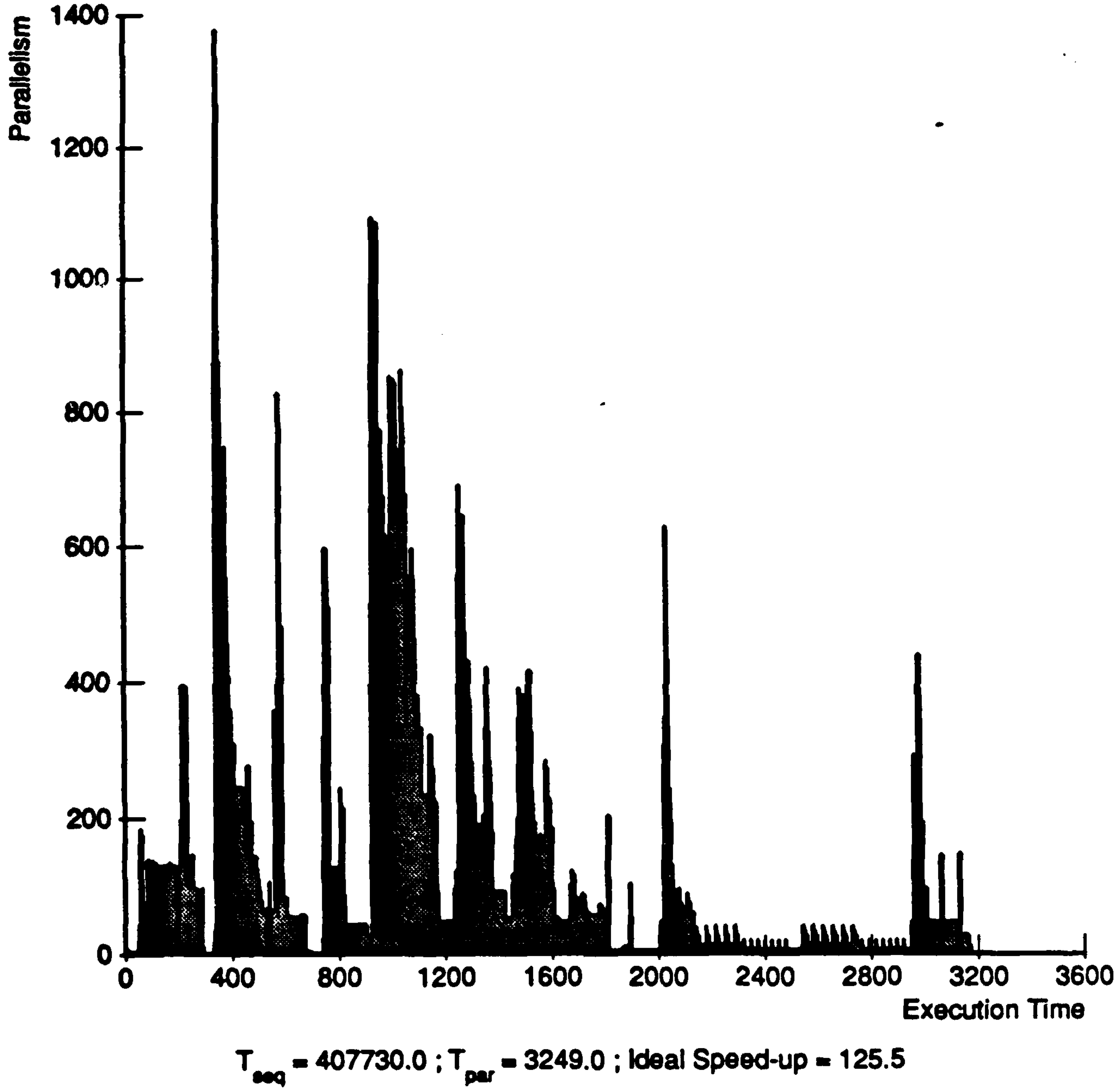


Figure 3: Parallelism profile for SIMPLE benchmark

on this task.

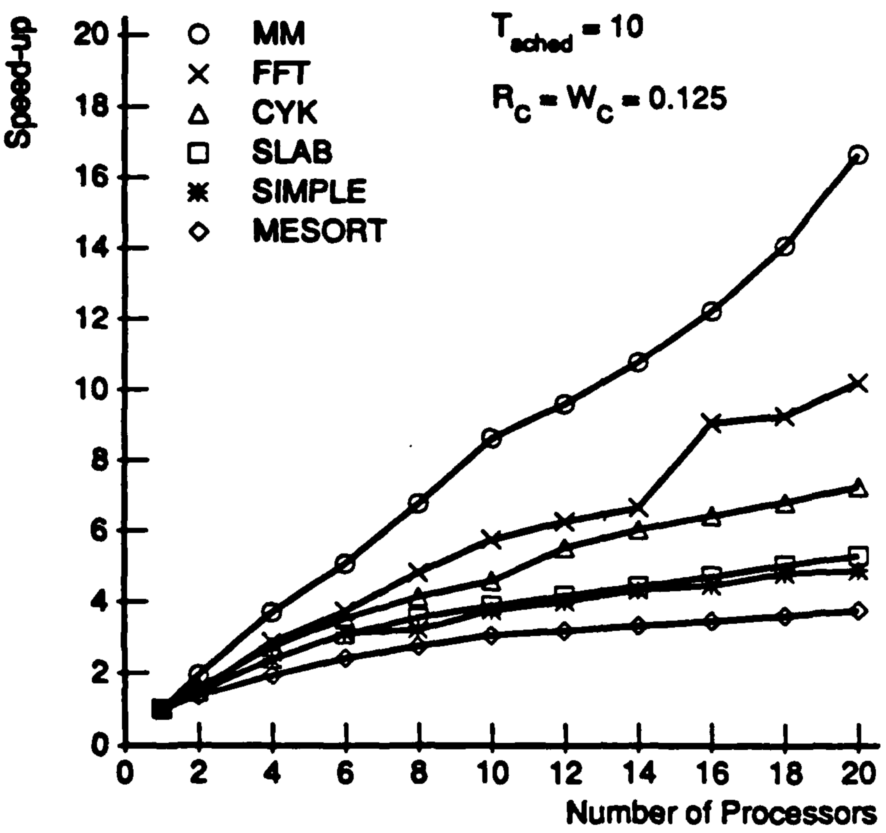


Figure 4: Typical speedup curves (simulated results)

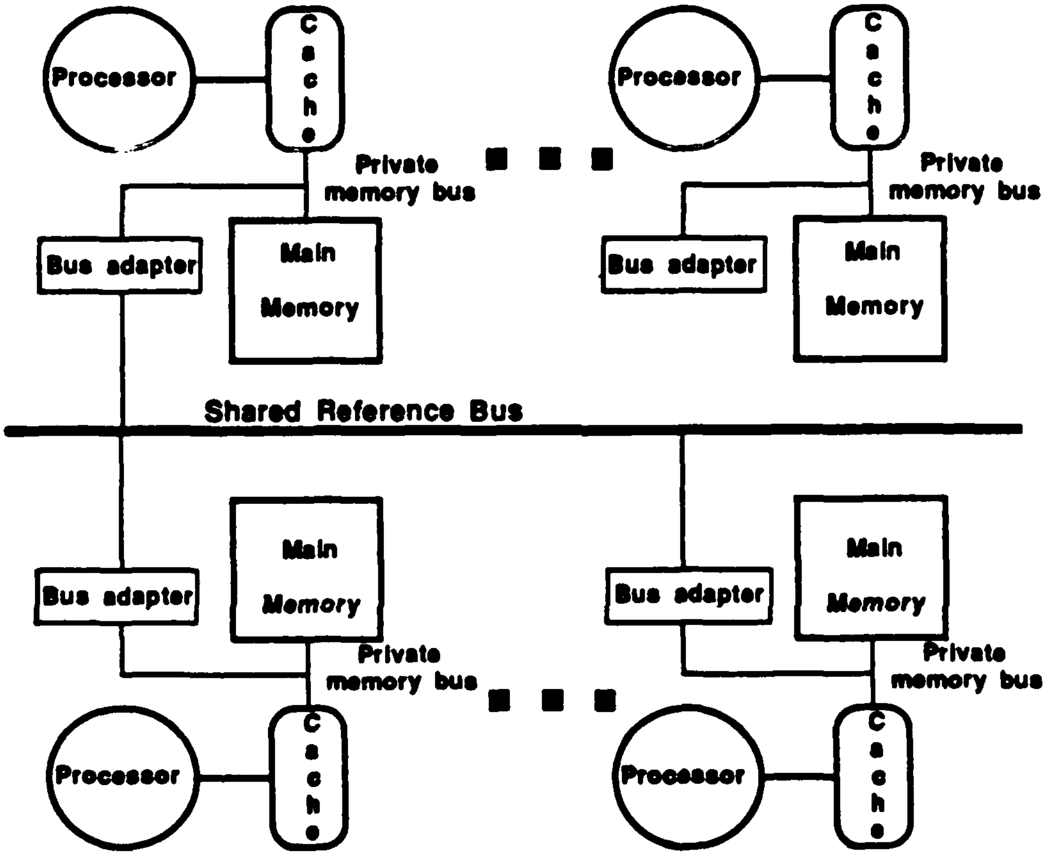
2.2.2 Shared Memory Multiprocessor Architecture

We have begun detailed design of a MIPS-X based shared memory multiprocessor. The architecture we are pursuing is called the distributed memory architecture and is shown in Figure 5. The primary advantage of this organization is that the bus need only handle shared references; cache misses are handled by the memory local to the processor. In a sense this organization allows the physical structure of the machine to reflect the logical structure. Our current plans for this machine include:

- Limited hardware support for maintaining cache coherency with software controlling and enforcing coherency; the absence of cache coherency not only reduces bus traffic, it also allows a faster processor, since arbitration at the cache is not needed,
- Hardware support for monitoring remote memory cache access to allow the operating system to migrate portions of memory if appropriate, and
- hardware measurement to allow bus activity to be monitored.

A detailed design of the processor unit is underway.

Figure 5: The Distributed Memory Architecture



2.2.3 MIPS-XMP Summary

Staff: S. Przybylski, J. Hennessy, M. Horowitz, M. Wing, P. Chow, A. Agarwal, J. Celoni, V. Sarkar, K. Gopinath, H. Davis, K. Gharachloo, S. Tjiang, J. Rose

Related Efforts: SPUR (Berkeley), Butterfly (BBN), Cosmic Cube (Caltech), RP3 (IBM)

References: [Sarkar 86b], [Sarkar 86a], [Hennessy 86]

2.3 High Speed Arithmetic

Two chips have been designed to test new ways of implementing high speed multiplication and division on silicon. The goal of this effort is to build high speed functional units that require modest silicon area, so they can be combined onto a floating point coprocessor. We are focusing on scalar operations, so our emphasis is on low latency and not simply throughput. To reduce the area requirements both chips use an iteration in time approach, where the data loops around a small array to produce the full output. The circuit design on the two chips is very different.

The division chips uses self-timed domino logic as the basic circuit structure. The chip consists of three radix 2 SRT division stages connected in a ring, a small amount of control logic for starting and stopping the iteration, and a set of shift-registers that accumulate the quotient bits. The three division stages form a ring oscillator that has the side effect of producing quotient bits. Each stage is completely self-timed and begins to evaluate as soon as the output of the previous stage is valid. The self-timing is accomplished by using dual-rail signal -- both the true and complement values are monotonic signals. This chip has been fabricated and tested. In 3u CMOS the chips run 23ns/quotient bit, while the 2u parts run 13ns/quotient bit. Besides it high speed, one of the key advantages of this approach is its small size. The entire 48 bit divider is only 6mm by 1.5mm.

The multiplication chips are currently in fabrication. They again use iteration, but this design uses a more conventional scheme with clocked latches. The chip implements a part of a Wallace Tree, and requires 7 clock cycles to complete a 64 by 64 multiply. The cycle delay should be roughly equal to the delay through two full adds and a register. Simulations indicate that this will be less that 18ns. To provide this high speed clock we have included a programmable clock generator on the chip. This should allow us to test the chip at speed using a low speed functional tester. A substantial fraction of the cycle time and area is used for the latches used in the design. We are now investigating methods to reduce the cost of the latches. The simplest method would be to go to a single phase clock, like the clocking style used in Crays machines.

Staff: T. Williams, R. Alverson, M. Santoro, M. Horowitz

References: [Williams 87]

3 Testing

3.1 Tester Design

3.1.1 The Data Generator/Receiver

The DGR is an attempt to use VLSI technology to make VLSI chips easier to test. It serves two functions: it acts as a small high speed vector memory, allowing burst vector rate of over 10MHz, and it acts as a configurable set of input/output pads optimized for driving the DUT (device under test). The current version of the DGR stores 256 vectors per pin, contains the electronics for 16 DUT pins, and is housed in a 84 pin PGA.

During this period we have received prototypes of the 2μ version of the DGR chips. These chips have been tested and are functional. The chips operate at over 16 MHz and after preliminary testing appear to be free of design errors. The yield on these parts was very poor, and none of the part were completely functional. We are using the mostly functional parts to try and debug the chip. This chip was resubmitted and we hope to have a better yield on the next run. Once the chips are debugged we will begin the design of a replacement tester for the Sun Kit 1 using the DGR chips.

3.1.2 High Speed Pin Drive

During this period we have designed a set of high-speed pin drive electronics. The circuit should be able to adjust output edges to about 1ns resolution, and run upto 30 MHz. The pins support all the standard formats RZ, RO, RT, RC, and NRZ and provide per pin control of the output and and input timing. The chip supports an adjustable output level, and a single analog input that defines the input theshold. This project is part of a integrated high-speed tester that we are designing. The layout of the pin electronics is finished and we are currently in final testing.

Staff: M. Horowitz, J. Gasbarro

References: [Miyamoto 87]

3.2 Testable CMOS Structures

Many BIST schemes are not suitable for designing large embedded PLAs because they cannot perform self-test at normal operating speed, and take too much area. Our BIST PLA solves the above problems by using a sequential parity checking technique to achieve high testing speed and novel circuit structures to minimize hardware overhead [Liu 86].

Pseudorandom testing techniques have been used to self-test ASICs. However, these techniques require lengthy fault simultion, only consider single stuck-at faults, and may not provide high fault coverage due to the existence of random-pattern-resistant faults. Our self-test structure achieves very high fault coverage with a centralized verification testing technique. This technique provides nearly 100% coverage for combinational faults (a superset of single stuck-at faults), and requires no fault simultaion [Liu 87]

Staff: D.L. Liu and E.J. McCluskey

4 The Fable Project

The Stanford FABLE project in semiconductor manufacturing science is working to change the nature of semiconductor manufacturing. The semiconductor industry typically uses mass production of DRAMs to debug and refine new processes before they are utilized for lower volume but higher profit products. The FABLE project is developing methodologies to permit the rapid development and engineering of new processes so that highly innovative products depending on new processes can be fielded without the necessity of prior mass production.

A key goal of the FABLE project is the *programmable factory*, an integrated system of manufacturing equipment, sensors, and computer hardware and software. Just as a programmer can rapidly modify and debug a complex computer program, so a process engineer should be able to modify and debug the complex processes which controls the manufacturing of semiconductors. To make the factory easier to program, we are developing a process CAD system.

A second key goal of the FABLE project is the *virtual factory*, a factory that can be run in simulation. Just as VLSI circuits are simulated before they are cast in silicon, so VLSI manufacturing processes should be simulated before they are run in the factory. We are developing a computer-based simulation system to simulate semiconductor manufacturing processes in their entirety, using knowledge about equipment, processes, materials, devices, and circuits to predict critical measurements of manufacturing performance such as yield, electrical performance, throughput, and equipment utilization.

The programmable factory and the virtual factory must be based on a large common knowledge base that captures knowledge about equipment, processes, materials, schedules and other aspects of semiconductor manufacturing. The different software systems needed to support process development tasks -- e.g., design, debugging, execution, data acquisition/interpretation, control, updating -- need to have access to similar knowledge. For efficiency of development and ease of maintenance, it is crucial that information about a piece of equipment, for example, not be encoded one way for process design and another way for process debugging. This argues for a declarative (as structure and statements) rather than procedural (as programs) representation of the knowledge. Specialized interpreters will exploit the common knowledge base for each task.

The Fable Project has made substantial progress in recent months. To demonstrate the potential of knowledge-based technology for semiconductor manufacturing applications, we have implemented three prototypes in process representation/editing, factory-level simulation, and communications network implementation. To attract students to manufacturing research, we ran a Stanford class called "Automation of Semiconductor Manufacturing. To make our results available to a larger audience, we have given numerous presentations and submitted several papers for publication.

4.0.1 Process Editor Prototype

The success of the Fable Project depends on our ability to represent and acquire a large body of knowledge about semiconductor manufacturing. A driving question for Fable is, "What does the automated factory need to know?". At a very general level, we know that the automated factory will need to know about processes, products, equipment, materials, facilities, costs, and safety, among other things.

Recently we have been investigating the more specific question, "What does the automated factory need to know about manufacturing processes?". As a tool to help us explore the representation of knowledge about processes, we have developed a prototype of a process editor. The process editor is implemented using KEE and Common Lisp on a TI Explorer workstation.

Our process editor provides two general capabilities. First, it provides a graphical facility for entering high-level descriptions of process flow. Second, it provides a forms-oriented interface for specifying detailed information about individual process steps. The result is a system that permits a process expert to interactively enter a description of a complete semiconductor fabrication process.

Presently, the graphical process editor only supports a strictly sequential process flow. We are now working to incorporate a multiplicity of programming control structures into the process editor, including conditionals, functional abstraction, and iteration.

The forms-oriented interface for editing descriptions of individual process steps supports the three Fable levels which have become standardized throughout the process specification community. These are:

- **Effect**: what physical effect the process step is expected to have on the wafer.
- **Treatment**: a machine-independent description of the environment the wafer will be placed in to achieve the specified effect.
- **Settings**: a machine-dependent description of the steps needed to accomplish the treatment.

In addition, we have begun to develop two addition "levels" to capture other information important to the process. These are:

- **Precondition**: the expected state of the wafer before it enters a process step.
- **Postcondition**: the expected state of the wafer after it completes a process step, including subtle side-effects.

The prototype has been highly valuable to our continuing research in process specification. By providing an interactive environment for process specification, it allows processing experts to clearly see the state of our work and to make their own contributions. Several processing engineers and graduate students specialized in particular types of processes have used the process editor to view and edit process descriptions. They have helped us elaborate the descriptions or specific process steps. More important, they have provided us with a long list of suggestions, many of which we plan to incorporate in our next version.

4.0.2 Factory-Level Simulation

To elaborate our vision of the virtual factory, we have implemented a prototype factory-level simulation system. This simulation is based on the *SimKit* knowledge-based discrete-even simulation system from IntelliCorp.

The current model consists of 29 servers, each server having an associated input queue and technician; 2 "sinks" for removing lots from the system, and one source for lots starts. Statistical variation in the model is introduced via generators, which can use one of a number of standard distributions made available by SimKit. A wide variety of data collection and monitoring capabilities exist for recording the value or state of any attribute of the system.

This system demonstrates the value of knowledge-based simulation tools. SimKit lets the user enter a description of the active elements of the factory and then to describe the behavior of each of the elements. SimKit also provides built-in facilities for graphically displaying the simulated activity and for collecting and analyzing simulation data.

We have multiple plans for extending this simulation. First, we plan to extend the specific simulation example to capture the entire Stanford 2-micron CMOS fabrication facility. Second, we plan to expand the detail of the factory-level simulation by incorporating aspects of equipment simulation. Third, we plan to explore the use of the SimKit interface as an interface to the real factory, not just the virtual factory. Finally, we plan to explore the use of parallelism in factory simulation, to substantially increase the performance of our simulation programs.

4.0.3 Equipment Communications

Work continued on the SECS interface under Unix 4.3 using the IP networking protocols and the Berkeley Unix socket mechnism. The equipment used as a test vehicle is a Varian 350D implanter equipped with a very complete SECS-I and II interface. A Unix interface (called a daemon) has been developed to allow arbitrary programs following simple rules to establish a connection with a specific piece of process equipment. This allows host computers with the correct capability (AI/Lisp, Smalltalk, or C++) to establish a direct link with the process equipment and perform tasks directly with the machine.

We have created an object-oriented programming environment using C++ to work with the SECS-I interface program. Object-oriented SECS-II programs written in C++ are able to access equipment via any machine running the SECS-I deamon. Such a programming environment is very well suited to SECS-II messages because of the formatting overhead in the messages. This style also hides the complexity of the process equipment communications and allows the application developer to concentrate on the operations to be performed. C++ has the added advantage of actually being compiled into ordinary C language used on all Unix systems and is therefore very portable.

We are expanding the C++ environment and the SECS-I deamon to allow arbitary test programs to be attached to the system. This will allow equipment simulation programs to be created and accessed transparently.

We are also beginning to create a compiler for SECS-II specification. The compiler will produce higher level SECS-II message objects for the C++ programs. This will allow complex application objects to be created automatically and will be used to create the "pseudo equipment" interface in a simulation program intended for connection in the manner described above.

4.0.4 Class: Automation of Semiconductor Manufacturing

Automation of Semiconductor Manufacturing (CS 412/EE 391) was led by Byron Davies and Jay M. Tenenbaum. It comprised ten 90-minute sessions, including 7 invited lectures. Speakers came from TI, Schlumberger, Fairchild, Stanford, and Carnegie-Mellon University. Lecture topics including industrial needs for CIM, CIM databases, process specification, testing and diagnostics, and expert systems for CIM.

Seven class projects involving ten students were carried out:

- Process specification and production simulation for VLSI manufacturing
- Numerical models for process diagnosis
- Equipment automation for reproducibility
- User-friendly interface for processing equipment
- Object-oriented SECS-II interface
- Ion implanter simulation
- Adaptive control of semiconductor manufacturing processes

4.0.5 IC-CIM Discussion List

A few months ago, we initiated a netwide mailing list, IC-CIM, to support discussion of computer-integrated manufacturing of integrated circuits. Since it started, IC-CIM has distributed about a dozen moderated messages, on a wide variety of topics. IC-CIM is seen as an important mechanism for disseminating information in the IC manufacturing research community.

4.0.6 Presentations and Papers

We have communicated the Fable vision and recent Fable progress in a number of different ways. We presented Fable to the SRC/Berkeley Workshop on System Architectures for Computer-Integrated Manufacturing. We presented Fable informally to audiences at Intel, Varian, and TI. We described Fable research to the Technical Advisory Board of the SRC. In addition, Fable research was described briefly by Prof. Jim Plummer in his invited talk at the Advanced Research in VLSI Conference at Stanford in March.

Members of the Fable'< Project have recently submitted four short Fable-related papers to the VLSI Technology Conference and the ElectroChemical Society Conference on Automated Manufacturing. These included:

- Davies, Leeke, and Saraswat: *Fable: Knowledge for Semiconductor Manufacturing*
- Leeke, Davies, and Saraswat: *The Virtual Fab Modeling System*
- Wood, Schenk, and Wijaya: *Object-Oriented Implementation of SECS I/II Protocols*
- Gardner and Davies: *Advanced Automation Techniques for Semiconductor Manufacturing Equipment*

Finally, at the Advanced Research in VLSI Conference, we conducted a two-hour workshop on process specification. This workshop was attended by researchers from Berkeley, CMU, MIT, and Stanford, as well as representatives from Intel, IBM, PROMIS Systems, and Lincoln Labs. This workshop explored industrial requirements for a process specification language, and investigated the similarities and differences between process specification formalisms being developed at each of the university sites.

Staff: B. Davies, M. Tenenbaum, P. Asente, L. Adams, E. Kirshenbaum, E. Wood.

Related Efforts: Hodges and Katz (Berkeley), McIlrath (MIT).

References: [Davies 87, Leeke 87, Fable87c, Gardner 87]

4.1 Microlithography

Work has been continuing in microlithography in the areas of e-beam mask making and direct write, optical and thin-film lithography, and defect inspection algorithms based on the template approach. Support work on Mebes and the Ultratech stepper for Center for Integrated Systems and other runs has been carried out.

4.1.1 Electron Beam Lithography

In this time period, 32 mask sets were generated on MEBES. In addition to CIS users, requests were from the physics department, as well as from the programs of Professors Pease, Quate Swanson, Harris, Gibbons, White, and Angell.

Work was started using capacitor structures to measure radiation damage to gate oxide structures by e-beam radiation. Initial results show some radiation induced damage, the effects of annealing on this damage is currently being investigated.

Wafers were patterned with 0.5μm lines and spaces for the task group on interconnections and contacts. These are to be used for selective W deposition in 0.5μm trenches to form both vias and interconnections.

Work to better quantify MEBES as a metrology tool has been started to continue the work completed under the 1/8 μm contract with Perkin-Elmer EBT. The initial application is to evaluate MEBES butting under standard writing conditions of three repeated scans for Shipley 2400 resist exposure, and those where the pattern stripe butting boundaries have been shifted in each scan to reduce by averaging the butting error at any one location. Initial results show an improvement in the resulting error, but that the metrology needs to be improved for errors less than 0.05 μm, 3 sigma. We feel that improvements can be achieved by using a differential backscatter detector and more uniform substrate resist topologies by an optimized resist process to gain better signal-to-noise ratios.

4.1.2 Defect Inspection

The prototype content-addressable memory chips received from MOSIS could not function due to errors in design data conversion. A corrected design (using 2 μm design rule) and a reduced version using 3 μm design rule were submitted again .

In addition to continuing the work on the template approach to pattern defect detection and classification and reporting results in various technical conferences, we have started investigating an alternative scheme for detection and classification. This scheme is based on the evaluation of *Euler numbers* of the local image and ite binary complement. It has been found that, for all typical pattern defect types (including random defecs and dimensional errors), simple rules can be generated for inspection. Computer simulations have shown a satisfactory defect coverage of this technique.

4.1.3 Langmuir-Blodgett Films

Electron-beam exposures were performed on both brassidic acid and cadmium brassidate films. In this experiment, 15 molecular layers of cadmium brassidate (for a total thickness of 308 angstroms) were deposited onto an aluminized silicon wafer using the LB technique. Brassidic acid samples were prepared by immersing the cadmium brassidate film in a 10^{-4}M HCl solution. These films were exposed on a SEM at 5 KV accelerating voltage, and the unexposed areas were dissolved in alcohol. The measured sensitivity were 7×10^{-4} and 2×10^{-3} coulombs/cm^2 for brassidic acid and cadmium brassidate, respectively, and the contrast γ values were 1.42 and 1.0, respectively.

The support for this project ended at the end of 1986.

4.1.4 Optical Lithography

Earlier work on the voting-lithography technique has emphasized on the aspect of amelioration of mask defects. There has been a concern about possible adverse impacts of such a multiple-field exposure scheme on the critical dimension (CD) unifiormity and overlay accuracy. However, recent studies in collaboration with Ultratech Stepper have demonstrated that these parameters obtained from voting exposures are consistenly superior to those from conventional exposures. Such improvements can be explained by the fact that the random components in CD variation and overlay error tend to be averaged through the superposition of multiple fields, thus resulting in tighter distributions.

We have received a donation from Ultartech Stepper to upgrade our stepper to a model 1000 system at no cost. This donation, valued at $159,000, can further enhance the photolithographic capabilities in the CIS. Installation of the upgrade has been completed in February.

Staff: R.F.W. Pease, D.H. Dameron, C.C. Fu, Soo-Ik Chae, Pierre Maccagno.

References: [Chae 86, Chae 87a, Wright 87, Chae 87b]

4.2 Processes, Devices, and Circuits

4.2.1 Dry Etching

During this period because of the move of the processing facility to the new CIS building, work in this area has focused on the design and building of a new experimental etch system, and on examining the effects of high rate resist stripping.

New Experimental Etch System

A commercial Drytek Model 100 etcher is being modified to have a wider range of operating conditions and to allow better monitoring of etch processes. This system should be a significant improve over the Plasmatherm etcher which had previously been used for our plasma diagnostics. Modifications include: operation in RIE or Plasma mode, RF and DC excitation, variable electrode spacing and area, chlorine based chemistries, a microbalance in the wafer electrode, additional optical windows and electrical feed-throughs, and circuitry for monitoring external currents and voltages to both electrodes. This system will greatly help our efforts in modeling and controlling etch equipment.

High Rate Resist Stripping

In the last year several companies have come out with single wafer high rate resist strippers. To understand the limitations of this equipment, we have investigated the effect of this process on very thin (< 15 nm) gate oxides, and have compared it to slower dry and wet strip processes. We have looked at minority carrier lifetime, fixed oxide charge, interface state density and oxide breakdown. Initial results indicate that the biggest problem is associated with the high temperatures (300 C) used to enhance the strip rate, in that at these temperatures diffusion of mobile ions out of the resist appears to deteriorate all parameters measured. We are presently looking at the effects of adding a halogen species to the oxygen plasma. Ion bombarment and UV radiation were not found to be significant problems for the processes investigated.

Staff: J. D. Shott, J. P. McVittie, K. C. Saraswat

Related Efforts: Oldham (Berkeley).

References: [McVittie 86, Sturm 87, Kao 87, McVittie 87, Ulacia 87a, Ulacia 87b]

References

[Agarwal 87] Agarwal, A., Chow, P., Horowitz, M., Acken, J., Salz, A., and Hennessy, J.
On-Chip Instruction Caches for High Performance Processors.
In *Advanced Research in VLSI*. Stanford University, Stanford, CA., March, 1987.

[Chae 86] S.I. Chae, J.T. Walker, D.H. Dameron, C.C. Fu, and J.D. Meindl.
Template-set approach to VLSI pattern inspection.
Automated Inspection and Measurement, SPIE J. 730, 1986.

[Chae 87a] S.I. Chae, J.T. Walker, C.C. Fu, D.H. Dameron, and R.F.W. Pease.
Template-set Approach to Defect Detection and Classification for VLSI Patterns.
In *Proceedings of the Topical Meeting on Machine Vision*. Optical Society of America, 1987.

[Chae 87b] S.I. Chae, J.T. Walker, C.C. Fu, and R.F.W. Pease.
Maskable Associative Memory Design for VLSI Pattern Inspection.
In *Technical Digest of 1987 VLSI Circuits Symposium*. May, 1987.

[Chow 85] Chow, P., Horowitz, M.
The MIPS-X Microprocessor.
In *Proc. WESCON 85*. IEEE, San Francisco, 1985.

[Chow 86] Chow, P.
MIPS-X Instruction Set and Programmer's Manual.
Technical Report 86-289, Stanford University, May, 1986.

[Chow 87] Chow, P., Horowitz, M.
Architectural Tradeoffs in the Design of MIPS-X.
In *submitted to 14th Intl. Sym. on Computer Architecture*. IEEE, 1987.

[Chu 86] Chu, C.Y., Horowitz, M.
Charge Sharing Models for MOS Circuits.
In *Intl. Conference on Computer-Aided Design*, pages 274-277. IEEE, Santa Clara, CA, November, 1986.

[Davies 87] B. Davies, S. Leeke, and K. Saraswat.
Fable: Knowledge for Semiconductor Manufacturing.
In *Proceedings of the Conference on Automated Manufacturing*. Electrochemical Society, 1987.
To be published.

[Gardner 87] D. Gardner and B. Davies.
Advanced Automation Techniques for Semiconductor Manufacturing Equipment.
In *Proceedings of the Conference on Automated Manufacturing*. Electrochemical Society, 1987.
To be published.

[Hennessy 86]	Hennessy, J. and Horowitz, M. *An Overview of the MIPS-XMP Project.* Technical Report 86-300, Computer Systems Laboratory, Stanford University, April, 1986.
[Horowitz 87]	Horowitz, M., et al. A 32b Microprocessor with On-Chip 2Kbyte Instruction Cache. In *Intl. Conference on Solid-State Circuits*, pages 30-31. IEEE, New York, NY, February, 1987.
[Kao 87]	D.B. Kao, J.P. McVittie, W.D. Nix, and K.C. Saraswat. Two-Dimensional Thermal Oxidation of Silicon - I. Experiments. *IEEE Transactions on Electron Devices* , 1987. To be published.
[Leeke 87]	S. Leeke, B. Davies, and K. Saraswat. The Virtual Fab Modeling System. In *Proceedings of the Conference on Automated Manufacturing*. Electrochemical Society, 1987. To be published.
[Liu 86]	Liu, D. and McCluskey, E. Design of CMOS VLSI Circuits for Testability. In *Proc. 1986 Custom Integrated Circuits Conference*. Rochester, NY, 1986.
[Liu 87]	Liu, D. and McCluskey, E. A VLSI CMOS Circuit Design Technique to Aid Test Generation. In *Proc. Phoenix Conf. on Computers and Communications*. Scottsdale, Az., 1987. To appear.
[McVittie 86]	J.P. McVittie, T.A. Lin, and A.J. Bariya. A New Method for Analyzing Thin Sidewall Inhibitor Layers. In *Photon, Beam and Plasma Stimulated Chemical Processes at Surfaces*. Materials Research Society, December, 1986. To be published.
[McVittie 87]	J.P. McVittie and T.A. Lin. Ion Suppression for Studying Etch Inhibitor Layers. In *Proceedings of the Sixth Symposium on Plasma Processing*. Electrochemical Society, 1987. To be published.
[Miyamoto 87]	Miyamoto, J., Horowitz, M. A Single Chip Functional Tester. In *Intl. Solid-State Circuits Conference*, pages 232-233. IEEE, New York, NY, February, 1987.
[Sarkar 86a]	Sarkar, V. and Hennessy, J. Optimal Granularity of Parallel Programs. 1986. Working paper.

[Sarkar 86b] Sarkar, V., Hennessy, J.
Compile-time Partitioning and Scheduling of Parallel Programs.
In *Sym. on Compiler Construction.* ACM, Palo Alto, Ca., June, 1986.

[Stark 87] Stark, D., Horowitz, M.
REDS: Resistance Extraction for Digital Simulation.
In *Design Automation Conference.* ACM, June, 1987.
to be published.

[Steenkiste 86] Steenkiste, P and Hennessy, J.
LISP on a Reduced-Instruction-Set Processor.
1986.
Working paper.

[Sturm 87] J.C. Sturm, J.P. McVittie, J.F. Gibbons and L. Pfieffer.
A Lateral Silicon-on-Insulator Bipolar Transistor with a Self-Aligned Base Contact.
IEEE Electron Device Letters EDL-8(3):104-106, March, 1987.

[Ulacia 87a] J. I. Ulacia and J.P. McVittie.
In-Situ Monitoring of Electrical Parameters for Dry Etching.
In *Proceedings of the 1987 MRS Symposium.* Materials Research Society, April, 1987.
To be published.

[Ulacia 87b] J. I. Ulacia, K. Sibbett, J.P. McVittie and R. Reichelderfer.
Gate Oxide Damage in Dry Photoresist Strippion Environments.
In *Proceedings of the 1987 MRS Symposium.* Materials Research Society, April, 1987.
To be published.

[Williams 87] Williams, T., Horowitz, M., Alverson, R., Yang, T.
A Self Timing SRT Division Chip.
In *Advanced Research in VLSI.* Stanford University, Stanford, CA, March, 1987.

[Wright 87] P. Wright, W. Loh, C.C. Fu, D.H. Dameron, and K. Saraswat.
Technology and Modeling of Submicron Contacts.
In *Proceedings of the 3^{rd} International Symposium on VLSI Technology, Systems, and Applications.* IEEE, 1987.

Publications

[Acken 87] Acken, J., Horowitz, M.
A Static RAM as a Fault Model Evaluator.
In *Custom Integrated Circuits Conference*. IEEE, May, 1987.
to be published.

[Agarwal 87] Agarwal, A., Chow, P., Horowitz, M., Acken, J., Salz, A., and Hennessy, J.
On-Chip Instruction Caches for High Performance Processors.
In *Advanced Research in VLSI*. Stanford University, Stanford, CA., March, 1987.

[Carpenter 87] Carpenter, C., Horowitz, M.
Generating Incremental VLSI Compaction Spacing Constraints.
In *Design Automation Conference*. ACM, June, 1987.
to be published.

[Chae 86] S.I. Chae, J.T. Walker, D.H. Dameron, C.C. Fu, and J.D. Meindl.
Template-set approach to VLSI pattern inspection.
Automated Inspection and Measurement, SPIE J. 730, 1986.

[Chae 87a] S.I. Chae, J.T. Walker, C.C. Fu, D.H. Dameron, and R.F.W. Pease.
Template-set Approach to Defect Detection and Classification for VLSI Patterns.
In *Proceedings of the Topical Meeting on Machine Vision*. Optical Society of America, 1987.

[Chae 87b] S.I. Chae, J.T. Walker, C.C. Fu, and R.F.W. Pease.
Maskable Associative Memory Design for VLSI Pattern Inspection.
In *Technical Digest of 1987 VLSI Circuits Symposium*. May, 1987.

[Chow 87] Chow, P., Horowitz, M.
Architectural Tradeoffs in the Design of MIPS-X.
In *submitted to 14th Intl. Sym. on Computet Architecture*. IEEE, 1987.

[Davies 87] B. Davies, S. Leeke, and K. Saraswat.
Fable: Knowledge for Semiconductor Manufacturing.
In *Proceedings of the Conference on Automated Manufacturing*. Electrochemical Society, 1987.
To be published.

[Gardner 87] D. Gardner and B. Davies.
Advanced Automation Techniques for Semiconductor Manufacturing Equipment.
In *Proceedings of the Conference on Automated Manufacturing*. Electrochemical Society, 1987.
To be published.

[Horowitz 87] Horowitz, M., et al.
A 32b Microprocessor with On-Chip 2Kbyte Instruction Cache.
In *Intl. Conference on Solid-State Circuits*, pages 30-31. IEEE, New York, NY, February, 1987.

[Kao 87] D.B. Kao, J.P. McVittie, W.D. Nix, and K.C. Saraswat.
Two-Dimensional Thermal Oxidation of Silicon - I. Experiments.
IEEE Transactions on Electron Devices , 1987.
To be published.

[Leeke 87] S. Leeke, B. Davies, and K. Saraswat.
The Virtual Fab Modeling System.
In *Proceedings of the Conference on Automated Manufacturing*. - Electrochemical Society, 1987.
To be published.

[Liu 86] Liu, D. and McCluskey, E.
Design of CMOS VLSI Circuits for Testability.
In *Proc. 1986 Custom Integrated Circuits Conference*. Rochester, NY, 1986.

[Liu 87] Liu, D. and McCluskey, E.
A VLSI CMOS Circuit Design Technique to Aid Test Generation.
In *Proc. Phoenix Conf. on Computers and Communications*. Scottsdale, Az., 1987.
To appear.

[McVittie 86] J.P. McVittie, T.A. Lin, and A.J. Bariya.
A New Method for Analyzing Thin Sidewall Inhibitor Layers.
In *Photon, Beam and Plasma Stimulated Chemical Processes at Surfaces*. Materials Research Society, December, 1986.
To be published.

[McVittie 87] J.P. McVittie and T.A. Lin.
Ion Suppression for Studying Etch Inhibitor Layers.
In *Proceedings of the Sixth Symposium on Plasma Processing*. Electrochemical Society, 1987.
To be published.

[Miyamoto 87] Miyamoto, J., Horowitz, M.
A Single Chip Functional Tester.
In *Intl. Solid-State Circuits Conference*, pages 232-233. IEEE, New York, NY, February, 1987.

[Schafft 87] H.A. Schafft, T.C. Staton, J. Mandel, and J.D. Shott.
Reproducibility of Electromigration Measurements.
IEEE Transactions on Electron Devices ED-34:673-681, March, 1987.

[Stark 87] Stark, D., Horowitz, M.
REDS: Resistance Extraction for Digital Simulation.
In *Design Automation Conference*. ACM, June, 1987.
to be published.

[Sturm 87] J.C. Sturm, J.P. McVittie, J.F. Gibbons and L. Pfieffer.
A Lateral Silicon-on-Insulator Bipolar Transistor with a Self-Aligned Base Contact.
IEEE Electron Device Letters EDL-8(3):104-106, March, 1987.

[Ulacia 87a] J. L Ulacia and J.P. McVittie.
In-Situ Monitoring of Electrical Parameters for Dry Etching.
In *Proceedings of the 1987 MRS Symposium*. Materials Research Society, April, 1987.
To be published.

[Ulacia 87b] J. L Ulacia, K. Sibbett, J.P. McVittie and R. Reichelderfer.
Gate Oxide Damage in Dry Photoresist Strippion Environments. -
In *Proceedings of the 1987 MRS Symposium*. Materials Research Society, April, 1987.
To be published.

[Williams 87] Williams, T., Horowitz, M., Alverson, R., Yang, T.
A Self Timing SRT Division Chip.
In *Advanced Research in VLSI*. Stanford University, Stanford, CA, March, 1987.

[Wood 87] E. Wood, H. Schenk, and A. Wijaya.
Object-Oriented Implementation of SECS I/II Protocols.
In *Proceedings of the Conference on Automated Manufacturing*. Electrochemical Society, 1987.
To be published.

[Wright 87] P. Wright, W. Loh, C.C. Fu, D.H. Dameron, and K. Saraswat.
Technology and Modeling of Submicron Contacts.
In *Proceedings of the 3rd International Symposium on VLSI Technology, Systems, and Applications*. IEEE, 1987.

END

10-87

DTIC